



POLITECNICO DI MILANO

FIDO WAS HERE

DESIGN AND IMPLEMENTATION OF MOBILE
APPLICATIONS PROJECT

Requirements Analysis and Specifications Document

Authors:

Sara GRILLO

Giulia VALCAMONICA

Professor:

Luciano BARESI

2020-2021

version 0

Contents

List of Figures	I
List of Tables	II
1 Introduction	1
1.1 Purpose	1
1.1.1 Goals	1
1.2 Scope	2
1.3 World, Machine and Phenomena	3
1.4 Glossary	4
1.4.1 Definitions	4
1.4.2 Acronyms	4
1.5 Document overview	4
2 Overall Description	5
2.1 Product perspective	5
2.2 Product functions	7
2.3 User characteristics	7
2.4 Assumptions, dependencies and constraints	8
3 Specific Requirements	9
3.1 External Interface Requirements	9
3.1.1 User Interfaces	9
3.1.2 Hardware Interfaces	14
3.1.3 Software Interfaces	14
3.1.4 Communication Interfaces	14
3.2 Functional Requirements	15
3.2.1 Users scenarios	15
3.2.2 User use cases	15
3.2.3 Requirements	21
3.3 Traceability scheme	22
3.4 Future Extensions	24

List of Figures

1 The World and the Machine	3
2 FidoWasHere class diagram	6
3 Login page, Mail Login/Registration page and Mail Registration page	11
4 Home page, Report page and Look for Fido page	11
5 Search Result page, View Report page	12
6 Retrieve My Report page, View My Report page	12
7 Statistics page	13
8 User's Use Case Diagram	15
9 Logged User's Use Case Diagram	16

List of Tables

Mail Sign Up	17
Mail Log in	17
FB Log in	18
User reports a Fido	18
User Looks for Fido	19
User Looks for Fido	19
User accesses statistics	20

1 Introduction

1.1 Purpose

The aim of this document is to describe what the application was intended for and highlight the functionalities required to fulfill all the users' needs. The document also provides critical information such as development, quality assurance and operations. Hence, it states which requirements and constraints the software needs to satisfy in order to correctly work.

FidoWasHere is an application that intends to provide users with the possibility to notify an encounter with a lost dog or cat by sending pictures and data about the Fido to the application database. Once that a Fido is reported by an user it can be searched by its owner.

It comes with an additional feature that allow the users to view statistics about the effectiveness of the application in terms of reported pets and found pets. It is also possible to visualize the reported Fidos' position, whose reports are still open, on the map.

1.1.1 Goals

The goals of FidoWasHere can be divided according to the main services it offers:

Basic service

- G1** Users can report a Fido.
- G2** Users are allowed to access data about a reported Fido.
- G3** Users can look for theirs lost Fidos.

Advanced service

- G4** Users can visualize the effectiveness of the app by accessing info about reported Fidos and found Fidos .
- G5** Users can visualize on a map the position of a reported Fido.

1.2 Scope

FidoWasHere is an application intended to be used by everyone for the sole purpose to help owners find their lost Fidos.

The applications grants two functionalities:

Basic service:

- Users will have the possibility to send reports through the application and FidoWasHere collects and stores them.
- Users can also look for their lost Fidos by filtering collected reports through data entries about the lost Fidos.

Advanced Service :

- FidoWasHere shows statistics about the effectiveness of the app and a map with the encountered location of the reported Fidos, whose reports are still open.

1.3 World, Machine and Phenomena

The terminology *the World and the Machine* refers to a way of describing the domain in which the FidoWasHere' software will operate. It distinguishes the problems and the phenomena the application will deal with into three areas. While the world contains the phenomena that happen in reality but are not observable by our system, the machine instead contains the phenomena which take place inside the system and are not observable from the outside. The intersection between these two sets of phenomena is the so called set of shared phenomena, which contains the phenomena that are controlled by the world and observed by the machine or controlled by the machine and observed by the world.

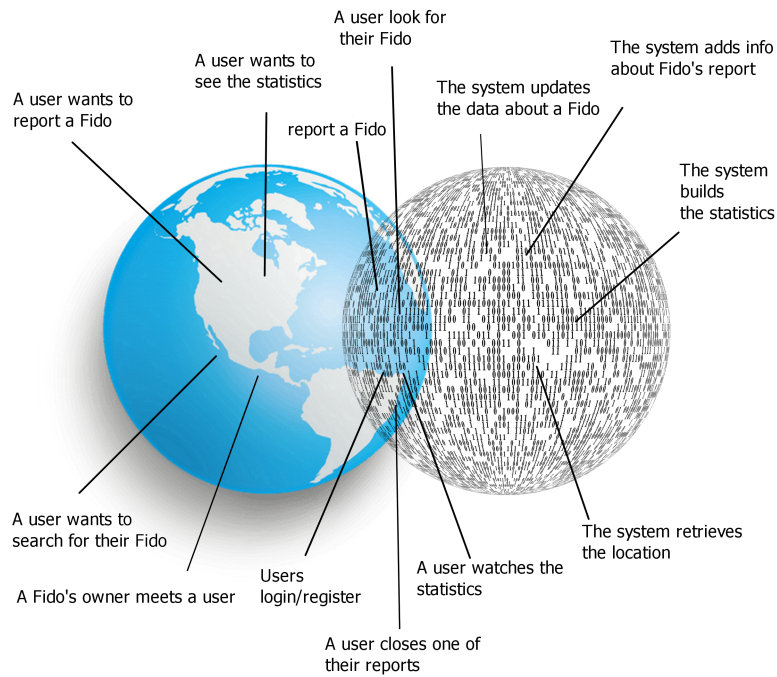


Figure 1: The World and the Machine

1.4 Glossary

1.4.1 Definitions

System: the FidoWasHere software we are to develop.

Android: one of the most common mobile devices' operating system.

User: person who has logged in and uses the system, also referred as Logged User.

Basic service: Users send pictures and details about a lost Fido that they've encountered.

Advanced Service: The System builds the statistics about the app effectiveness and visualize the reported Fidos on the map.

Fido: Dog or Cat

1.4.2 Acronyms

RASD: Requirements Analysis and Specification Document.

UI: User Interface.

GPS: Global Position System.

DB: Database.

UML: Unified Modeling Language.

FB: Facebook.

1.5 Document overview

1. **Introduction:** it provides an overview of this entire document, product goals and the description of the world and its phenomena
2. **Overall description:** it describes general factors that affect the product providing the background for system requirements, such as the domain assumptions
3. **Specific requirements:** it contains all system's functional and non-functional requirements and also the scenarios of the system with the use case diagrams
4. **Conclusions:** it contains final considerations on the project

2 Overall Description

2.1 Product perspective

The system that is going to be developed is not based on any existing software and will be newly invented. On the idea that users will be able to send pictures of Fidos they find, it is considered that the system will have a database for storing the pictures together with information about the report that are sent through the smartphone application.

The database it's also necessary to offer the advanced service, it will be inquired by the system to retrieve data about the stored report. Once that FidoWasHere has these data, it is possible to use them to visualize on the map information about the report and build statistics.

When a Fido is found, its report will be closed and its row in the application DB will not be delete, but the Fido will be marked as found, in this way the system can build statistics that can be view by the users on the application.

On Figure 2 the UML class diagram points out that users use the smartphone application to report about lost Fidos, look for lost Fidos, and to view statistics.

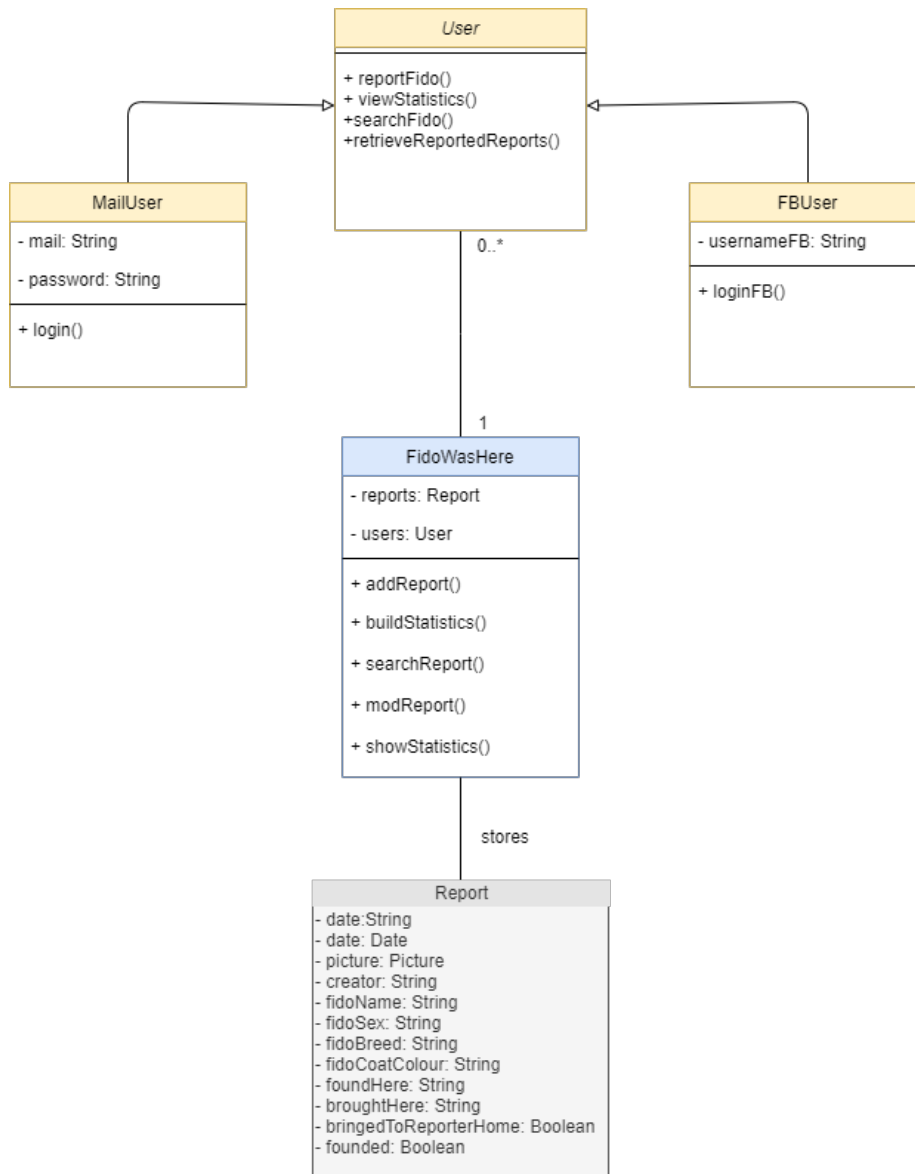


Figure 2: FidoWasHere class diagram

2.2 Product functions

Once the goals of the system have been identified, it is possible to underline the main functions that FidoWasHere will perform. Since the system can be divided into two main services, the functionalities are categorized in the same way.

Basic service functions The system needs specific information about the reported Fido in order to correctly store the data. The user has to take a picture of the Fido. Time and date are also saved considering the moment in which the user sends the report. These data are collected and saved so that they can be accessed from the database by other users through searching operations. A report can be closed only by the user who wrote it, to make this operation easier a user can access a page with all the active reports that they've made. Regarding the user's privacy, when sending a report along with the data about the encounter in the Fido's file there is also the mail or the FB account used for the login in order to be reached by the Fido's owner. Although the user's exact position isn't necessarily needed to complete the report's data the user can share their GPS location instead of typing it to identify the location on the map, in this case the software is going to ask permission to retrieve and store the user's location.

Advanced Service functions The system provides a map where it's possible to view the reported Fidos whose reports are still open, the map view by default is centered accordingly to the user's position, if the user has given the permission to access their location, otherwise the systems uses a placement for the coordinates. The Advanced Service also provides statistics regarding reported Fidos and found Fidos to measure the effectiveness of the application .

2.3 User characteristics

The application services are directed only to users. It is possible to classify users into those who will use the application to report or to find a Fido and those that will use the statistics although this classification is not inflexible, in fact a user can do all these actions without being asked for any particular requirements excluding functionalities' requirements.

Logged users Users that have downloaded and installed the application and successfully login, based on the way they've logged in into the system they can be *FBUsers* or *MailUsers*. The smartphone must have a stable internet connection. The logged user can be classified into *Reporting User*, *Searching users* and *Statistics User*

Reporting users Logged users that want to send a report about a Fido, they must have an image precedent taken of the Fido or a photo camera installed on their phone to take the photo. Reporting users have to grant the access to gallery and camera in order to send a report.

Searching users Logged users that want to look for their Fidos, they don't require any other requirement.

Statistics users Logged users that want to know the report position of the reported lost Fidos and access to information about reported and found Fidos.

2.4 Assumptions, dependencies and constraints

Domain assumptions clarify what it is expected from the external environment and set the limits of the software-driven machine so as to stay in the domain of interest.

- DA1** GPS position is supposed to be accurate.
- DA2** Users have an active and working internet connection while using the application.
- DA3** Users don't have any software on their smartphone that could change the GPS position.
- DA4** Users send only photos about lost Fido's.
- DA5** All the photos contain a single subject.
- DA6** Users compile the reports with correct data entries about the Fidos they encountered.
- DA7** Users send reports only about encountered Fidos and in the same day of the encounter.
- DA8** A report is close by its creator only after that the creator itself is reached by the Fido's owner and the Fido's retrieved by the latter.
- DA9** The users don't change current time or data when sending a report.
- DA10** Registering users use a valid and existing mail.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Using the interfaces of the system users on the application can:

- Login/Register with E-mail or log-in with a FB account
- Report a lost Fido
- Search for a lost Fidos
- View their active reports
- View statistics about the effectiveness of the app and active reports on a map

The mobile application is designed for common users who do not need to have competence of technological skills. Therefore it has an easy to use and minimal UI. Starting from the login page(Figure 3), the users can choose which login method suits them. Once logged in the user accesses the home page (Figure 4), where they can choose which section to access between four options: *Report*, *Look for Fido*, *My Reports* and *Statistics*.

Report:

The user takes (or loads) a picture about a lost Fido, complete the requested fields, and send it to the system (Figure 4). The completed fields should help to clarify the position and the condition of the lost Fido and simplify the process of *Look for Fido*.

Here are some explanation of the requested fields for the report:

- *Type of pet* : drop down menu, it requires to specify if the encountered Fido's a cat or a dog.
- *Pet Breed*: drop down menu, offers different categories of breed from which the user can choose, if the breed of the Fido is not clear the drop menu contains an *unknown* value that can be selected.
- *Coat colour*: drop down menu, it contains a list of colour, only one colour can be selected, if the Fido's coat has multiple colours the drop menu has a *mixed* value that can be selected.
- *Sex*: drop down menu, this field requires to specify the sex of the encountered Fido, if it's not possible to be verified the user can select the *unknown* value in the list.
- *Name*: text field where the user can write the name of the Fido if it's possible otherwise it can be let empty
- *Add photo*: by pressing the camera button the user can decide to take a picture of the Fido or to load it from their gallery.

- *found here* and *brought to*: by pressing these two fields the user opens a map where they can search for the desired addresses, if instead they press on the GPS buttons the system takes the actual position of the user (previous consent is required) and complete the correspondent field with that value.

Look for Fido

The user can look their Fido (Figure 4), for doing it the application requires to complete some fields about the lost Fido, these fields are the same as the ones contained into the report's screen with the exception of the field *Lost on*. When a user presses the field a calendar opens and user selects the date when they've lost their Fido.

After pressing the search button the user obtains some possible reports regarding their Fido. By pressing on one of them a page is open and it's possible to view the data about the report itself (Figure 5).

My Reports

The user can look up their active sent reports. By pressing on a sent report a page with all details of that report is opened. On this page you also see (Figure 6) a button *close* and if the button is pressed the user closes the report and the Fido is considered found by the system.

Statistics

The user has a map that they can move, zoom in and out or type in the name of an area of which they would like to see how many active report are in a zone. By default the map view is centered accordingly with the user's position. Under the map there are some information about the usage of the app (Figure 7).

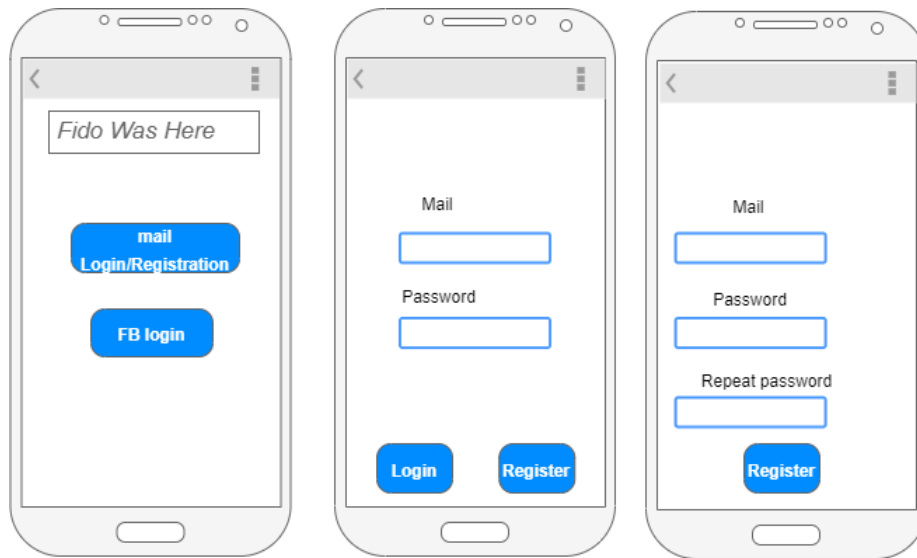


Figure 3: Login page, Mail Login/Registration page and Mail Registration page

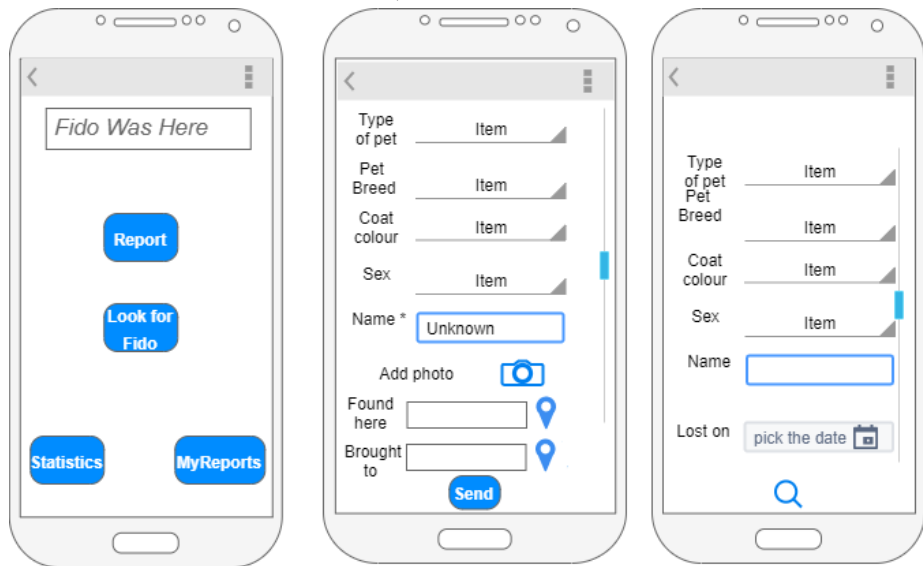


Figure 4: Home page, Report page and Look for Fido page

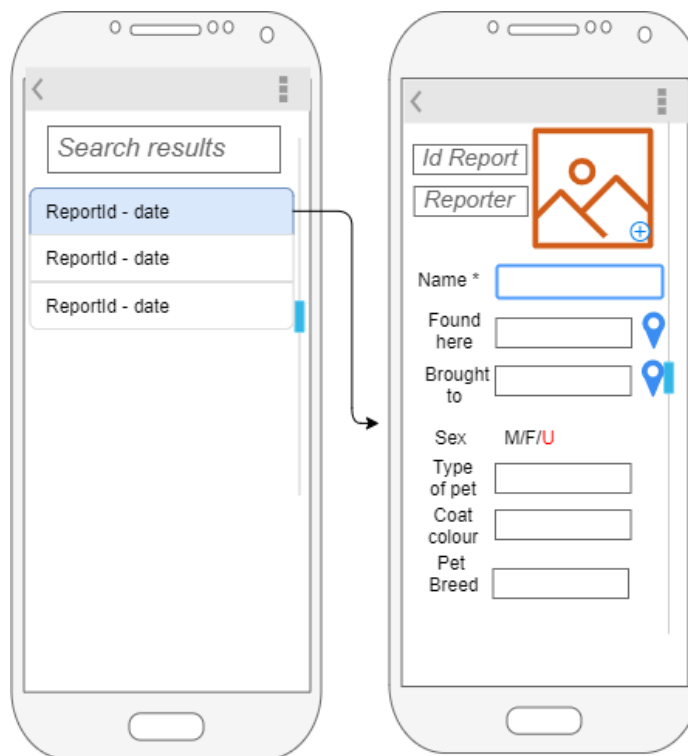


Figure 5: Search Result page, View Report page

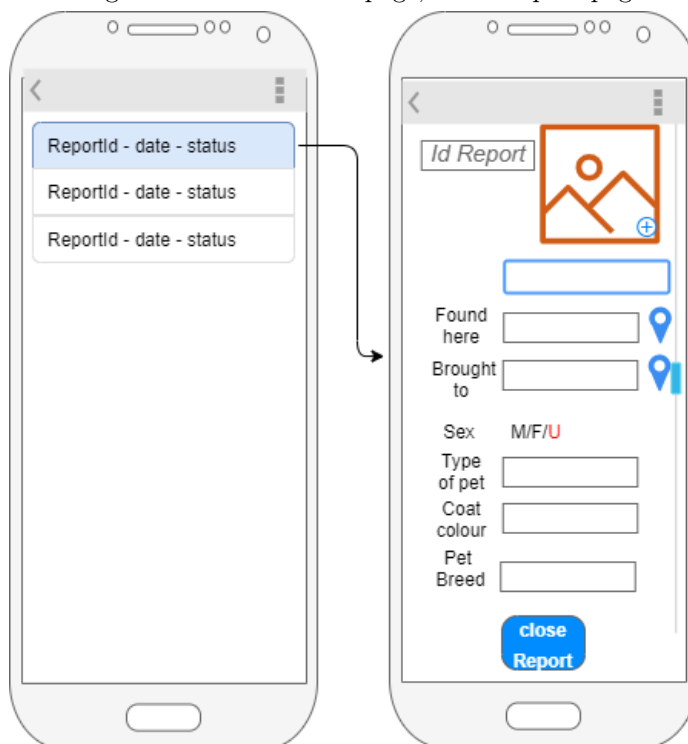


Figure 6: Retrieve My Report page, View My Report page

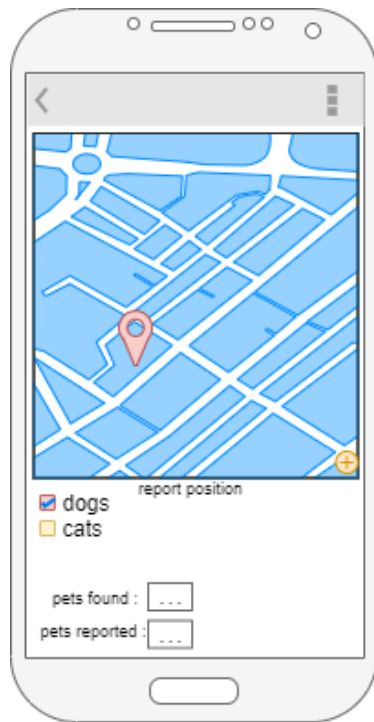


Figure 7: Statistics page

3.1.2 Hardware Interfaces

The hardware needed considering the entire system can be divided into the two main parts.

- The user app that needs these hardware components to run properly:
 - smartphone to run the application
 - stable internet connection for accessing the online services
 - GPS module to retrieve the position of the user to use the *Statistic* functionality
- The back-end machines that allow the system to perform its services:
 - databases to store all the data

3.1.3 Software Interfaces

Google Maps

This service allows the user to see the active reports on a map and it's also used when compiling a report.

Facebook

The service is used to obtain the Facebook user profile and email, in order to permit authentication through FB and to provide a way to be contact by other users.

3.1.4 Communication Interfaces

The communication between the system and those service providers is crucial because the system depends on those services to perform its functions. Given this, the systems communication interface must be either WiFi or Mobile Data (2G, 3G or 4G), and the service providers communication interface can be of any type, as long as it ensures that they are connected to the internet and this for the simple reason that the application will not allow the user to send any report if they are not connected.

3.2 Functional Requirements

3.2.1 Users scenarios

- S1** Mario has spotted multiple times during the month the same cat in Viale Romagna. The cat initially was clean and well fed but now its coat is dirty and the cat has visibly lost some weight, so he decides to report it as a lost Fido through his FidoWasHere app. Once he has taken a picture of the cat and filled in the information required on the fields in the app he then sends the report.
- S2** Luca has lost his dog Nemo during a walk in the park. He has redone the same track multiple times in order to find him, but without any luck. He then tries to search for his dog using the FidoWasHere application. He fills the required fields with information about his Fido on the Look for Fido section and the system responds with a list of reports with similar criterias reported by other users. Among those Luca finds Nemo and contacts the reporting user that has taken the dog to his house. Now Nemo can go back to his house and the reporting user can close the report.
- S3** Maria and Leah are very curious and want to know how many dogs are actually lost and reported within their location, Maria opens her FidoWasHere application to acknowledge this information. She accepts to share her GPS information and selects to view only the reports regarding dogs.

3.2.2 User use cases

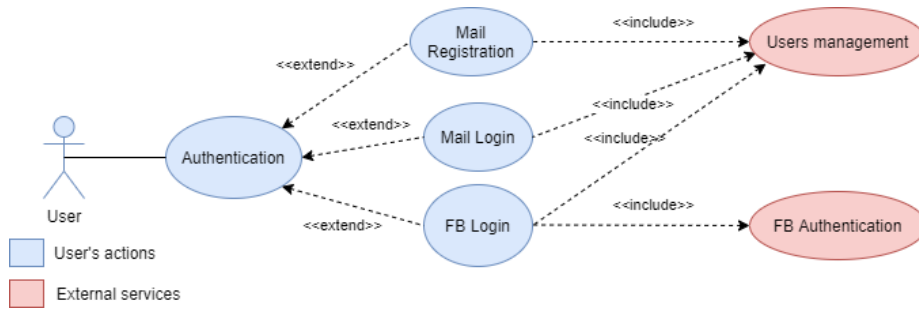


Figure 8: User's Use Case Diagram



Figure 9: Logged User's Use Case Diagram

Name	Users Mail Registration
Actors	User
Entry Condition	The user must have successfully installed the application.
Events Flow	<ol style="list-style-type: none"> 1. The User accesses the <i>Login/Sign-up</i> section. 2. The User presses on the <i>Sign-up</i> option. 3. User chooses the credentials that uniquely identifies them. 4. The user types a password. 5. The user confirm the password. 6. Confirmation is sent by clicking the <i>Sign-up</i> button.
Exit Condition	The User sent all the data for the registration and their row is created into the application's DB.
Exceptions	<ul style="list-style-type: none"> • User inserts an already existing mail. • The two password field are different. <p>The system rejects the registration .</p>

Name	Users Mail Login
Actors	User
Entry Condition	The user must have successfully installed the application.
Events Flow	<ol style="list-style-type: none"> 1. The User accesses the <i>Login/Sign-up</i> section. 2. User inserts the credentials that uniquely identify them. 3. The user types the password. 4. Login request is sent by clicking the <i>Login</i> button.
Exit Condition	The User is logged in
Exceptions	<ul style="list-style-type: none"> • User has sent an mail that is not in the DB. • User has written a wrong password. <p>The system rejects the login request .</p>

Name	Users FB Login
Actors	User
Entry Condition	The user must have successfully installed the application.
Events Flow	<ol style="list-style-type: none"> 1. The User accesses the <i>FB Login</i> section. 2. The User accesses through the application with their FB profile.
Exit Condition	The User is logged in
Exceptions	<ul style="list-style-type: none"> • User has discarded the FB prompt to login. • User is unable to access their FB profile. <p>The system rejects the login request .</p>

Name	User reports a lost Fido
Actors	User
Entry Condition	The user must have successfully installed the application on the smartphone and must be logged in.
Events Flow	<ol style="list-style-type: none"> 1. The user opens the application on the smartphone. 2. The user navigates in the <i>Report</i> section. 3. The user takes a picture through the application or load a picture from the gallery. 4. The user provides the data regarding the reported Fido. 5. The report is sent by clicking the <i>send</i> button.
Exit Condition	The system receives the violation data sent by the user.
Exceptions	<ul style="list-style-type: none"> • The user has not complete all the required fields. • The user does not have an internet connection. <p>In both cases the application informs the user that the option of reporting a violation is not available.</p>

Name	User searches their lost Fido
Actors	User
Entry Condition	The user must have successfully installed the application on the smartphone and must be logged in.
Events Flow	<ol style="list-style-type: none"> 1. The user opens the application on the smartphone. 2. The user navigates in the <i>Look for Fido</i> section. 3. The user provides the data regarding their Fido. 4. The search data are sent by clicking the <i>send</i> button. 5. The user visualize the resulting reports
Exit Condition	The system provides the reports accordingly with the data sent by the user.
Exceptions	<ul style="list-style-type: none"> • The user has not complete all the required fields. • There aren't any corresponding report. <p>In the first case the user is asked to fill all the field and retry. In the second case the user is informed that there aren't any information about the inserted field.</p>

Name	User closes a report
Actors	User
Entry Condition	The user must have successfully installed the application on the smartphone and must be logged in.
Events Flow	<ol style="list-style-type: none"> 1. The user opens the application on the smartphone. 2. The user navigates in the <i>My reports</i> section. 3. The user selects the desired report. 4. The user closes the report by pressing <i>close</i> button.
Exit Condition	The system updates its DB with the new status about the report.
Exceptions	<ul style="list-style-type: none"> • The user hasn't done any report. • There is an error while sending the new status about the report. <p>In both cases the system doesn't update the report status and asks the user to retry.</p>

Name	User accesses Statistics
Actors	User
Entry Condition	The user must have successfully installed the application on the smartphone and must be logged in.
Events Flow	<ol style="list-style-type: none">1. The user opens the application on the smartphone.2. The user navigates in the <i>Statistics</i> section.
Exit Condition	The application shows the information.
Exceptions	<ul style="list-style-type: none">• There are not statistics to show.• The user user has not activated the GPS.• The user has not shared the GPS coordinates <p>In all the cases the application informs the user that the option is not available.</p>

3.2.3 Requirements

The following requirements are derived in order to fulfill the specified goals.

G1 Users can report a Fido.

R1 The system can retrieve street address from the data GPS coordinates and store it

R2 The system shall allow application users to add reports

R3 The system shall store data about a violation into the application's DB

G2 Users are allowed to access data about a reported Fido

R4 The system shows reports information

G3 Users can look for their lost Fidos

R5 The system updates reports status

R6 The system filters the reports accordingly with the users' choices

R7 The system can updates reports' status

G4 User can visualize the effectiveness of the app by accessing info about reported Fidos and found Fidos

R8 The system builds statistics out of the information retrieved on reported Fidos

G5 Users can visualize on a map the position of the reported Fido

R9 The system updates reports status

R10 The system retrieves street address from the data GPS coordinates and store it

R11 The system uses a map service

3.3 Traceability scheme

To keep track of the relation between goals, requirements, domain assumptions and use cases a traceability scheme is provided.

G1 Users can report a Fido.

R1 The system can retrieve street address from the data GPS coordinates and store it

R2 The system shall allow application users to add reports

R3 The system shall store data about a violation into the application's DB

DA1 GPS position is supposed to be accurate

DA2 Users have an active and working internet connection while using the application

DA3 Users don't have any software on their smartphone that could change the GPS position.

DA4 Users send only photos about lost cats or lost dogs

DA5 All the photos contain a single subject

DA6 Users compile the reports with correct data entries about Fidos position

DA7 Users send reports only about encountered Fidos and in the same day of the encounter

DA9 The user doesn't change current time or data when sending a report

G2 Users are allowed to access data about a reported Fido

R4 The system shows reports information

DA2 Users have an active and working internet connection while using the application

DA7 Users send reports only about encountered Fidos and in the same day of the encounter

DA8 A report is close by its creator only after that the creator itself is reached by the Fido's owner and the Fido's retrieved by the latter

G3 Users can look for their lost Fidos

R5 The system updates reports status

R6 The system filters the reports accordingly with the users' choices

R7 The system can updates reports' status

DA2 Users have an active and working internet connection while using the application

DA7 Users send reports only about encountered Fidos and in the same day of the encounter

- DA8** A report is close by its creator only after that the creator itself is reached by the Fido's owner and the Fido's retrieved by the latter
- G4** User can visualize the effectiveness of the app by accessing info about reported Fidos and found Fidos
 - R8** The system builds statistics out of the information retrieved on reported Fidos
 - DA2** Users have an active and working internet connection while using the application
- G5** Users can visualize on a map the position of the reported Fido
 - R9** The system updates reports status
 - R10** The system retrieves street address from the data GPS coordinates and store it
 - R11** The system uses a map service
 - DA1** GPS position is supposed to be accurate
 - DA2** Users have an active and working internet connection while using the application
 - DA3** Users don't have any software on their smartphone that could change the GPS position.

3.4 Future Extensions

In a possible extension users will be able to register missing reports about their lost Fidos and reporting users will be notified by the system when sending a report if some existing missing reports corresponds to the sent report, if present the user can pick up the missing report corresponding to their report and binding them, after that the bound is created the application will notify the user that have posted the missing report. In another possible extension the system will allow more types of Fidos such as rabbit, reptiles and birds, the development of the application is going to consider this last option and it will try to be as general as possible in the term of Fidos.